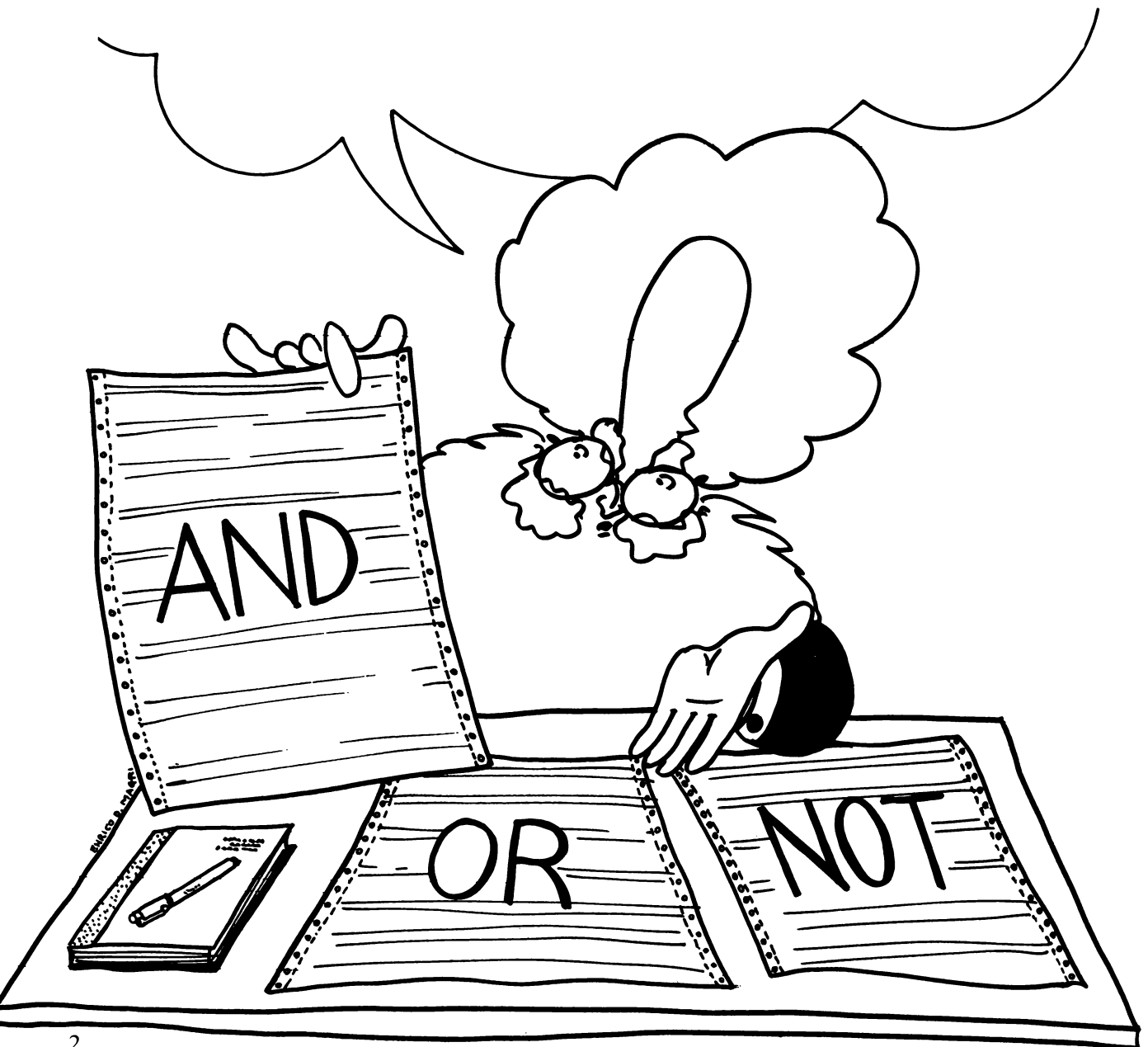
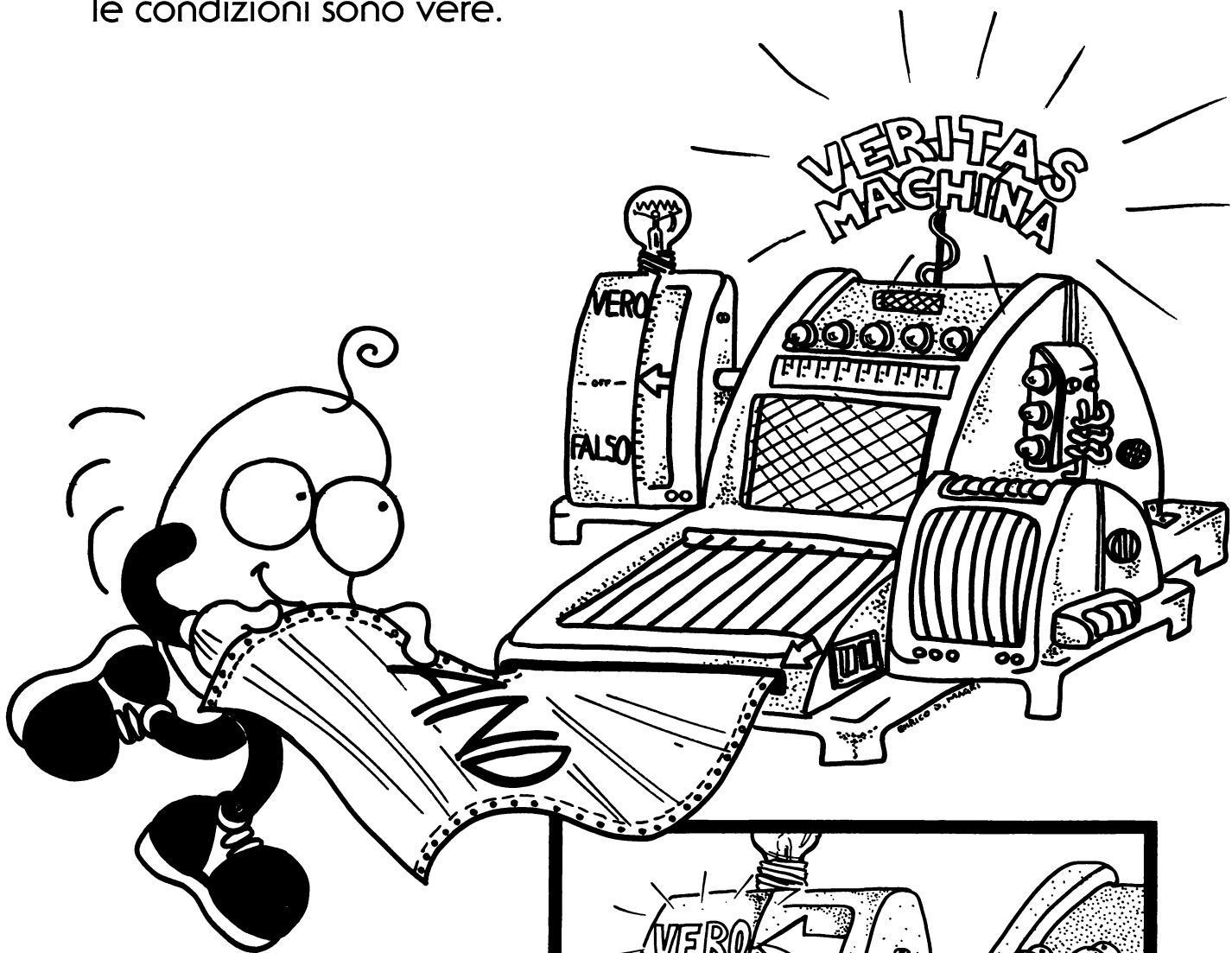
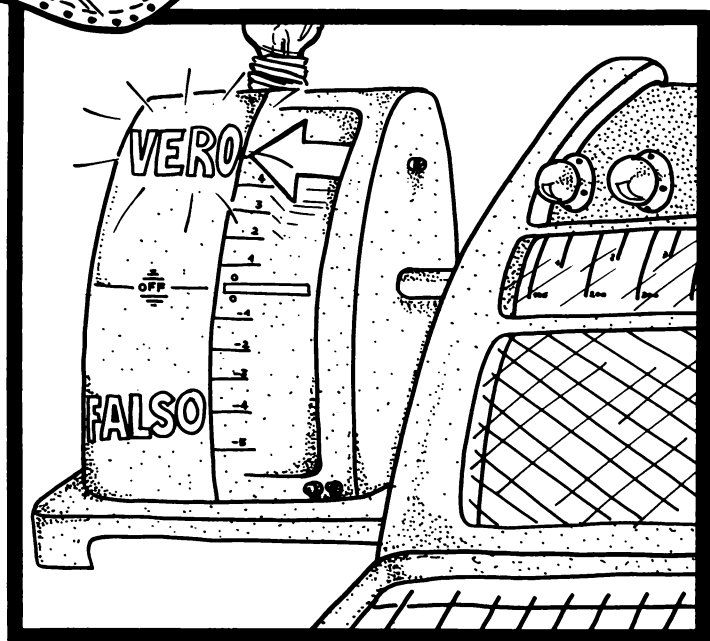
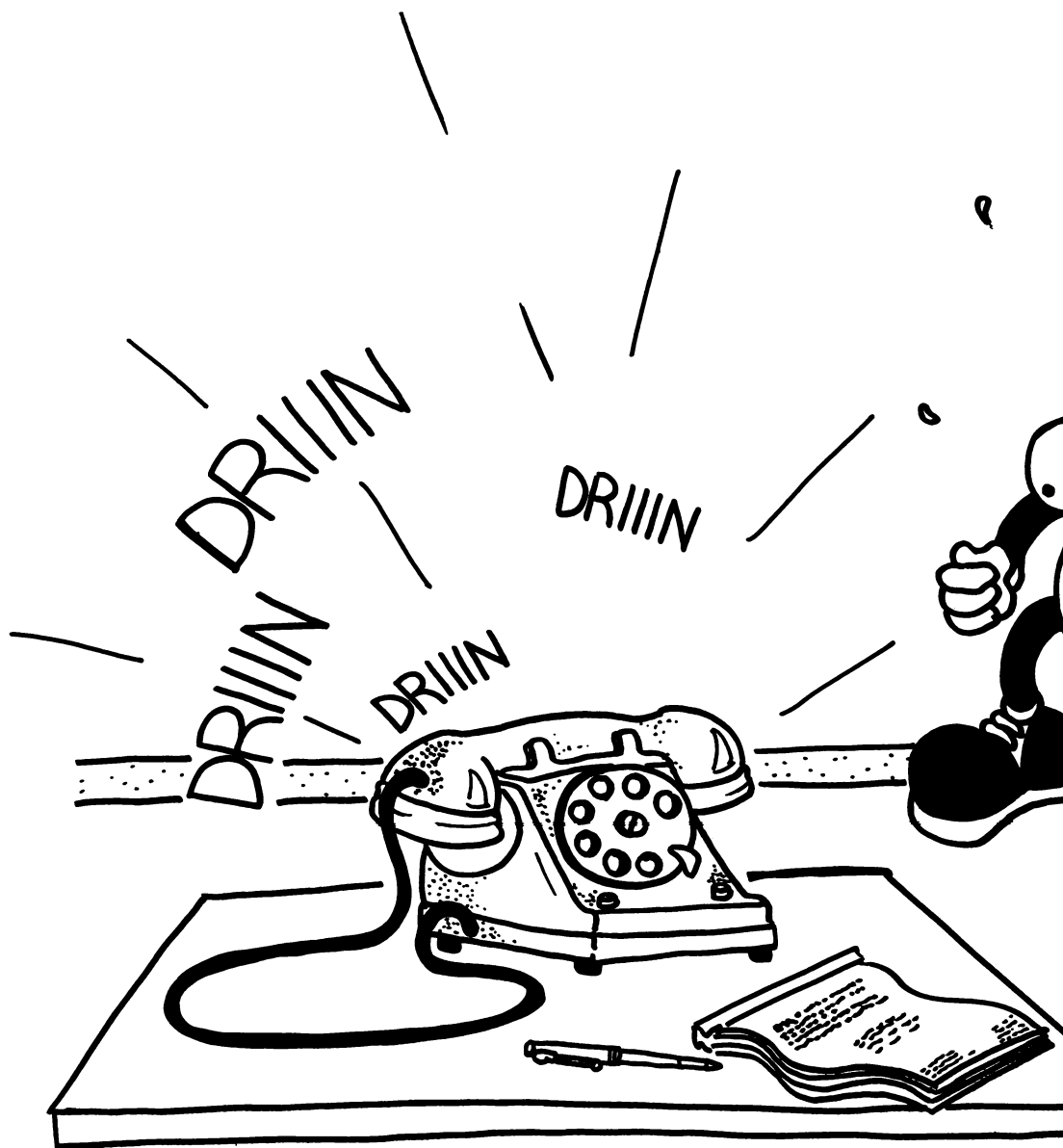
CONOSCERE IL COMPUTER DIRETTAMENTE DAL COMPUTER

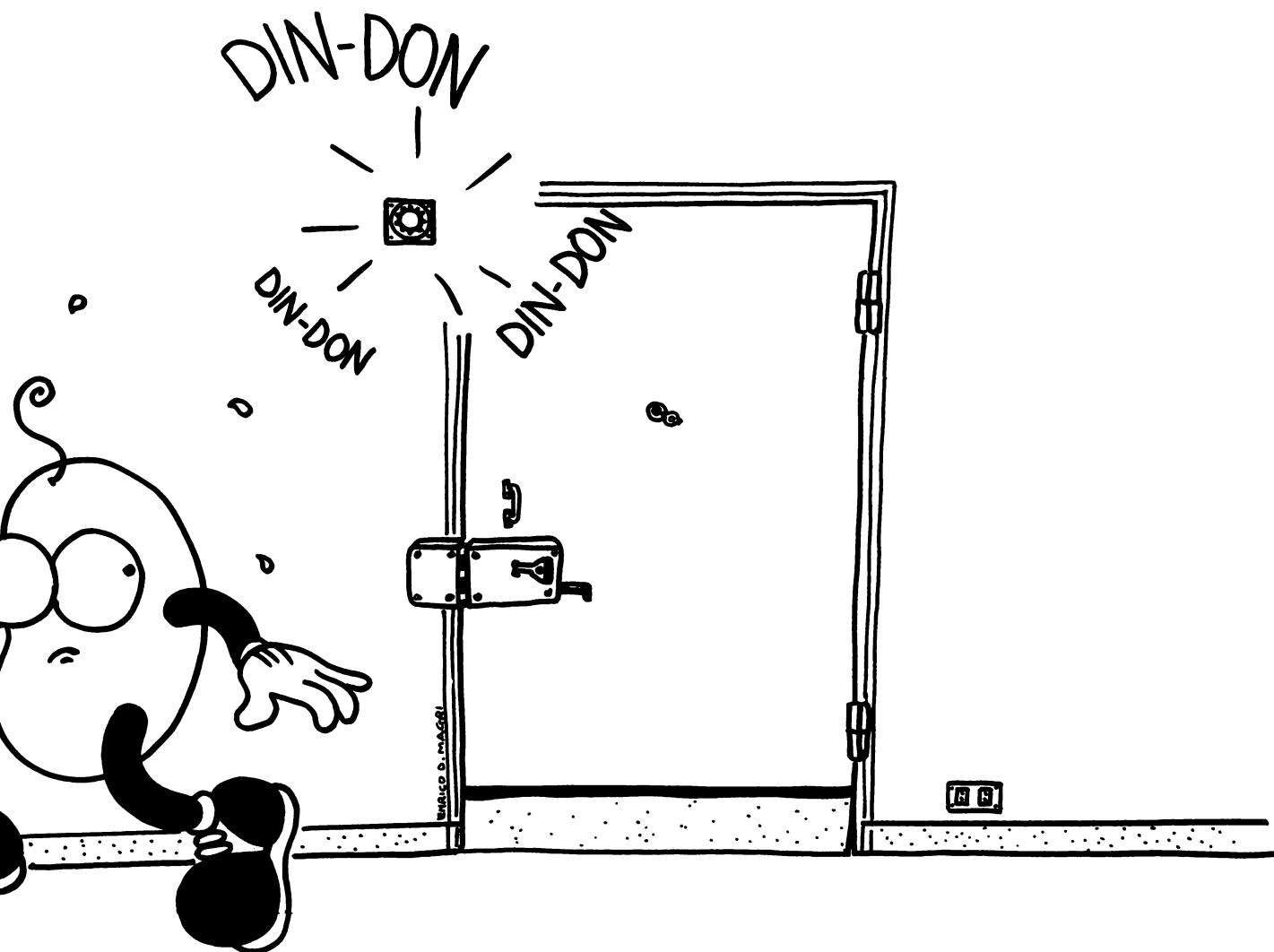
per Commodore Vic20 e 64

Beatrice d'Este

Gli operatori logici usati insieme all'istruzione IF, ti permettono di verificare all'interno del programma più condizioni contemporaneamente.
Gli operatori logici sono AND, OR, NOT, ma quelli usati solitamente sono l'AND e l'OR, chiamati anche rispettivamente **prodotto logico** (*) e **somma logica** (+).
Devi usare gli operatori logici nella forma: IF (condiz. 1) (op. logico) (condiz. 2) THEN (una istruzione).

Naturalmente puoi usare anche più di un operatore logico e di conseguenza più di due condizioni.
Usando l'AND il test risulterà vero, cioè verificato, se entrambi le condizioni sono vere.

Con l'OR, invece, il test risulterà vero se almeno una delle condizioni è vera.
Con il NOT il test risulterà vero se la condizione è falsa, cioè non verificata.
Ma il NOT non si usa mai.

In pratica l'AND svolge la funzione della congiunzione E, come nella frase:
SE SUONA IL TELEFONO **E** BUSSANO ALLA PORTA ALLORA RISPONDO.
Mentre l'OR svolge la funzione della congiunzione OPPURE, come nella frase:
SE SUONA IL TELEFONO **OPPURE** BUSSANO ALLA PORTA ALLORA RISPONDO.
Puoi anche scrivere invece dell'AND o dell'OR semplicemente i segni * e +, in tal caso dovrai racchiudere ogni condizione tra le parentesi.

Ad esempio:

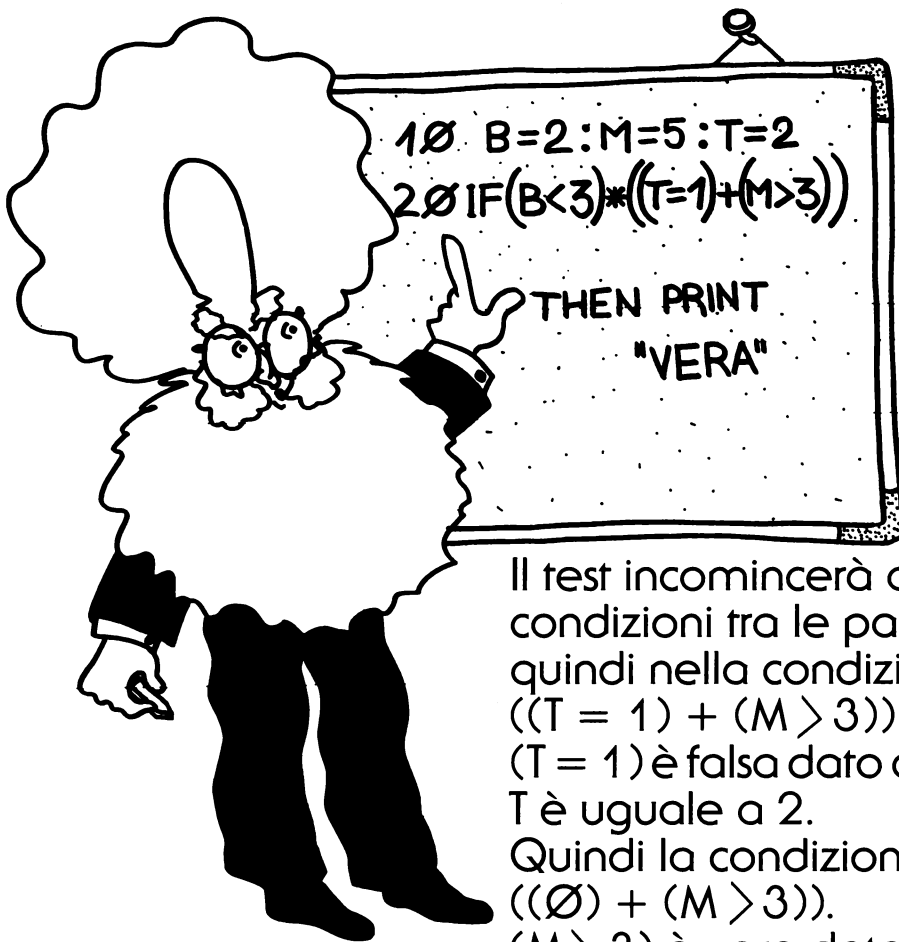
```
1Ø INPUT"NUMERO (1—1Ø)"; N
2Ø IF (N < 1) + (N > 1Ø) THEN 1Ø
```

Oppure scriverai il nome stesso degli operatori AND e OR.
Ad esempio:

```
1Ø INPUT"NUMERO (1 — 1Ø)"; N
2Ø IF (N < 1 OR N > 1Ø THEN 1Ø
```

Nei due esempi appena visti la linea 1Ø chiederà un numero tra 1 e 1Ø.
La 2Ø, nel caso il numero inserito è minore di 1 oppure è maggiore di 1Ø, cioè se non è accettabile, provvederà a far ripetere l'inserimento.

Naturalmente puoi forzare la priorità di una condizione usando le parentesi come fai nelle assegnazioni.
Per sapere se un test con più condizioni è vero, ti conviene sostituire ogni condizione con un 1 se è vera o con uno Ø se è falsa.

Quindi in:

```
1Ø B = 2: M = 5: T = 2
2Ø IF (B < 3) * ((T = 1) + (M > 3)) THEN PRINT "VERA".
```

Il test incomincerà ad analizzare le condizioni tra le parentesi più interne, quindi nella condizione posta in linea 2Ø ((T = 1) + (M > 3)) risulterà che:
(T = 1) è falsa dato che invece in linea 1Ø T è uguale a 2.
Quindi la condizione diventerà (B < 3) * ((Ø) + (M > 3)).
(M > 3) è vera dato che M = 5.
Perciò: (B < 3) * ((Ø) + (1)).

Facendo l'OR tra le due condizioni, dato che almeno una è vera, ((Ø) + (1)) risulterà vera.
A questo punto la condizione si ridurrà a: (B < 3) * (1).
Anche la condizione (B < 3) è vera dato che B è uguale a 2 e quindi risulterà: (1) * (1).
Alla fine, facendo l'AND, dato che entrambe sono vere, il test risulterà vero.

Listato dell'esercizio: ESEMPIO PRATICO PER L'UTILIZZO DELL'AND E DELL'OR

```
10 PRINT "Premi S o N"
20 GET A$
30 IF(A$<>"S")*(A$<>"N")THEN 20
40 PRINT"Inserimento accettato"
```

Listato dell'esercizio: COMPLETA IL LISTATO

```
10 DIM|*|(15),B$(15)
20 INPUT"QUANTI NOMI(1-15)";|*|
30 IFN>15 THEN|*|
40 FORK=1TON
50 INPUT"NOME";|*|
60 NEXTK
70 H=1
80 FOR|*|TON
90 FORK=1TON
100 IFA$(I)<A$(K)|*|H=H+1
110 NEXTK
120 IFB$(H)<>""THEN|*|:GOTO120
130 B$(H)=A$(I):|*|:NEXTI
140 FORK=NTO1STEP|*|
150 PRINTB$(K)
160 |*|K
```

PROGRAMMIAMO INSIEME (VIC 20)

```
10 POKE36879,127
20 PRINTCHR$(144)
30 INPUT"PAROLA";P$
40 L=LEN(P$)
50 FORK=LTO1STEP-1
60 C$=C$+MID$(P$,K,1)
70 NEXTK
80 PRINT"PAROLA INVERTITA:":PRINTC$
```

PROGRAMMIAMO INSIEME (CBM 64)

```
10 POKE53280,7:POKE53281,7
20 PRINTCHR$(144)
30 INPUT"PAROLA";P$
40 L=LEN(P$)
50 FORK=LTO1STEP-1
60 C$=c$+MID$(P$,K,1)
70 NEXTK
80 PRINT"PAROLA INVERTITA:":PRINTC$
```

Soluzione dell'esercizio "COMPLETA IL LISTATO"

(apparso nella lezione n. 18)

```
10 DIMD(12)
20 PRINT"ISTOGRAMMI"
30 INPUT"QUANTI DATI(1-12)";D
40 IFD=0THEN30
50 IFD>12THEN30
60 FORK=1TOD
70 INPUT"DATO";D(K)
80 IFD(K)>ATHENA=D(K)
90 NEXTK
100 FORK=1TOD
110 GOSUB200
120 NEXTK:END
200 L=INT((D(K)/A)*21)
210 FORW=1TOL
220 PRINT"■";
230 NEXTW:PRINT:RETURN
```